

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

* * * * *

DIGITAL DATA STORAGE SYSTEMS,
COMPUTERS, AND DATA VERIFICATION
METHODS

* * * * *

INVENTORS

Bennett J. Groeneveld

Wayne E. Austad

Stuart C. Walsh

Catherine A. Herring

ATTORNEY'S DOCKET NO. LIT-PI-484

EL 465682689

1 **CROSS REFERENCE TO RELATED APPLICATIONS**

2 This application claims priority from U.S. Provisional Application
3 Serial No. 60/128,605, filed April 8, 1999, titled "Digital Signature
4 System With Non-Repudiation For Relational Databases", having attorney
5 docket number LIT-PI-484, and incorporated herein by reference.

6
7 **TECHNICAL FIELD**

8 The invention provides digital data storage systems, computers, and
9 data verification methods.

10
11 **BACKGROUND OF THE INVENTION**

12 Digital signatures have been provided for non-repudiation wherein
13 the user is associated with a given set of data and the signer can not
14 deny participation with the signature. A digital signature associates a
15 digital or numerical code with a set of electronic data. The code is
16 generated using a private key that uniquely identifies the person that
17 is approving the contents of the data.

18 To create a digital signature, an encryption process, such as DES
19 or Triple DES for example, is utilized with a private key to encrypt a
20 hash of the data set. The private key is associated with a given user.
21 Thereafter, a public key which corresponds to the private key is utilized
22 to decrypt the encrypted data to arrive at the original data set.

23 Digital signatures provide security benefits of identification and
24 authentication plus data integrity to arrive at non-repudiation. The

1 identity of the signing person of a transaction is known and can be
2 proven to a third party. The signature is linked to the user. The
3 signature is also linked to the data being signed such that if the data
4 is changed, the signature is invalidated. For non-repudiation, the signing
5 party can not deny having signed the transaction inasmuch as the
6 signature is linked to the user and the data.

7 Previous paper-oriented tasks such as vendor payment were
8 processed by routing paper around the organization. Approvals were
9 hand-written signatures on paper. However, the aims of arriving at
10 substantially paperless initiatives and improving the practice of obtaining
11 authorization have resulted in wider acceptance and usage of digital
12 signatures.

13 There exists a need to provide improved storage and verification
14 systems utilizing digital signatures.

15 BRIEF DESCRIPTION OF THE DRAWINGS

16 Preferred embodiments of the invention are described below with
17 reference to the following accompanying drawings.

18 Fig. 1 is an isometric view of an exemplary digital data storage
19 and verification system according to one embodiment of the present
20 invention.

21 Fig. 2 is a functional block diagram of the system of Fig. 1.

22 Fig. 3 is a flow chart illustrating an exemplary methodology for
23 creating a digital signature.
24

1 Fig. 4 is a flow chart illustrating further details of an exemplary
2 methodology for creating a digital signature.

3 Fig. 5 is a flow chart illustrating an exemplary methodology of
4 flat verification of a digital signature.

5 Fig. 6 is a flow chart illustrating an exemplary methodology of
6 cross-verification of a digital signature.

7 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

8 This disclosure of the invention is submitted in furtherance of the
9 constitutional purposes of the U.S. Patent Laws "to promote the
10 progress of science and useful arts" (Article 1, Section 8).

11 According to a first aspect of the invention, a computer includes
12 an interface adapted to couple with a dynamic database; and processing
13 circuitry configured to provide a first hash from digital data stored
14 within a portion of the dynamic database at an initial moment in time,
15 to provide a second hash from digital data stored within the portion of
16 the dynamic database at a subsequent moment in time, and to compare
17 the first hash and the second hash.

18 A second aspect of the invention provides a system comprising:
19 storage circuitry configured to store digital data at least some of which
20 dynamically changes with respect to time, and to store a digital
21 signature generated using digital data stored within the storage circuitry
22 at an initial moment in time; and processing circuitry configured to
23 provide a first hash from the digital signature, and to provide a second
24 hash from digital data stored within the storage circuitry at a

1 subsequent moment in time and corresponding to the digital data of the
2 digital signature, and to compare the first hash and the second hash.

3 Another aspect of the invention provides a digital data system
4 comprising: storage circuitry configured to store digital data at least
5 some of which dynamically changes with respect to time; and processing
6 circuitry configured to compare a first hash and a second hash, wherein
7 the first hash corresponds to digital data stored within the storage
8 circuitry at an initial moment in time and the second hash corresponds
9 to digital data stored within the storage circuitry at a subsequent
10 moment in time.

11 According to another aspect, a digital data storage system
12 comprises: a dynamic database containing a plurality of tables
13 individually configured to store digital data; a snapshot database
14 configured to store a snapshot of digital data retrieved from at least
15 one table of the dynamic database at an initial moment in time; an
16 approval database configured to store a digital signature of the
17 snapshot; and a client configured to provide the digital signature from
18 the snapshot, to provide a first hash from the snapshot, to provide a
19 second hash from the digital signature, to compare the first hash and
20 the second hash, to provide a third hash from data stored within the
21 at least one table of the dynamic database at a subsequent moment in
22 time, and to compare the third hash and the second hash.

23 Another aspect provides a data verification method comprising:
24 storing digital data at an initial moment in time within a dynamic

1 database; providing a first hash of the digital data stored at the initial
2 moment in time; providing a second hash of digital data within the
3 dynamic database at a subsequent moment in time; and comparing the
4 first hash and the second hash.

5 Yet another aspect provides a data verification method comprising:
6 providing digital data at an initial moment in time within a portion of
7 a dynamic database; storing a static form of the digital data stored at
8 the initial moment in time within a static database; providing a digital
9 signature using the static form of the digital data; providing a first hash
10 of the digital data stored at the initial moment in time using the digital
11 signature; providing a second hash of the digital data stored at the
12 initial moment in time using the static form of the digital data;
13 comparing the first hash and the second hash; providing a third hash
14 of digital data stored within the portion of the dynamic database at a
15 subsequent moment in time; and comparing the first hash and the third
16 hash.

17 Referring to Fig. 1, an exemplary configuration of a digital data
18 storage and verification system 10 is shown. The depicted configuration
19 of system 10 includes a client terminal 12 coupled with a certificate
20 authority (CA) server 14, an application server 16, and a card reading
21 device of a private key management system 18. Client terminal 12 is
22 implemented as a workstation personal computer (PC) configured to
23 communicate with servers 14, 16 and the card reading device in the
24 described arrangement.

1 Client terminal 12 comprises processing circuitry 13 such as a
2 microprocessor configured to execute software instructions to perform
3 digital data storage and verification operations described herein. An
4 exemplary microprocessor is a Pentium III™ processor available from
5 Intel Corporation. Such storage and verification operations may be
6 implemented during a workflow wherein data is routed to one or more
7 people or signing entities for approval and signature.

8 Certificate authority server 14 is utilized to implement public key
9 infrastructure (PKI) operations in the described system arrangement.
10 For example, certificate authority server 14 issues new digital certificates
11 to a user (e.g., signer using client terminal 12) and digitally signs the
12 user's certificate with the CA's private key so the user's certificate can
13 be publically verified using the CA's public key at a later point in
14 time.

15 Client terminal 12 accesses certificate authority server 14 to
16 generate a private key and public key pair which enables a user via
17 client terminal 12 to digitally sign data sets creating digital signatures.
18 Typically, only a user has access to a given generated private key while
19 the certificate authority server 14 stores the public key which
20 corresponds to the private key. Server 14 also stores user identification
21 information enabling subsequent retrieval of the public key corresponding
22 to the particular user.

23 Numerous users may utilize client terminal 12 in some
24 embodiments. Alternatively, numerous client terminals 12 (not shown)

are provided within system 10. Such plural users input respective identification information which identifies themselves and their appropriate public key(s). The public keys may be used for verification of digital signatures previously signed by the users at subsequent points in time. Individual users have unique private keys and the associated public key(s) are stored within certificate authority server 14.

System 10 is described with a client/server Internet style architecture. Such is exemplary and other arrangements are possible. The depicted configuration could be implemented on a single machine or distributed between multiple servers without changes to basic operations and data verification, retrieval and storage.

Referring to Fig. 2, additional details of system 10 are shown. Private key management system 18 is coupled with client terminal 12. Private key management system 18 may be implemented at least in part as a smart card system wherein private keys and copies of the user certificates are stored on corresponding smart cards (e.g., Litronic tokens, Fortezza cards, etc.) of the user. Alternatively, private keys are stored within an Internet browser database (e.g., Explorer, Netscape, etc.) resident upon client terminal 12. Other implementations of private key management system 18 are possible. Any PKCS token can be utilized to manage private keys in the described embodiment. One exemplary arrangement of system 10 utilizes Litronic smart cards, RSA/MD5 algorithms, and internally managed certificate authority from Xcert International, Inc.

Private key management system 18 permits unlocking of private keys using pass-phrase or user identification information (also referred to as personal identification information) of corresponding users. Private key management system 18 is independent of specific encryption algorithms used to generate public-private key pairs at the time of certificate issuance (e.g., DSA, RSA, PGP, ECC, etc.) or other hash/ciphers chosen as part of encryption (DES, Triple-DES, MD5, SHA).

Certificate authority server 14 includes a certificate database 20 and certificate authority 22. Certificate database 20 includes public keys and certificates for current and past certificate holders (users) to be used when verifying digitally signed objects at subsequent points in time.

Certificate authority 22 comprises instruction code resident upon certificate authority server 14 configured to implement operations relative to user certificates and user public keys, including storage, retrieval and signing of such certificates and public keys including encryption of same with a private key corresponding to the respective certificate authority server 14.

Although the certificate authority server 14 typically does not have access to private keys of users, server 14 can revoke or periodically expire issued certificates to prevent new signatures from being created by given certificates. Currently available solutions for those wishing to internally implement their own certificate authority to manage their own certificates are available from Xcert International, Inc. and Entrust

Technologies, for example. Alternatively, certificates may be rented from commercially recognized certificate authorities such as VeriSign, GTE, Thawt, AT&T, etc.

Less secure certificate management systems, such as PGP-based signature systems may be utilized wherein public keys are stored in public accessible databases. Regardless of the desired level of security, individual certificate authority issued certificates and individual public keys have a unique fingerprint identification (user identification information) which correlates them to a specific user. The certificates and public keys are signed by a mutually trusted validating party such as certificate authority 22 using its own private key, or in the case of PGP systems, other users can sign (vouch for) another user's certificate.

Application server 16 comprises storage circuitry 17 configured to store and retrieve static and dynamic digital data as described in further detail below. In the depicted arrangement, application server 16 comprises a snapshot broker 30, and storage circuitry 17 comprising a snapshot database 32, an approval table database 34, and a dynamic database 36.

In an exemplary arrangement, application server 16 implements client-server database system operations with respect to client terminal 12. Client terminal 12 is coupled via an interface 24 (such as a network interface card) with application server 16 and components within application server 16. Application server 16 presents data to client terminal 12 through graphical forms or a hierarchical tree of forms, for

1 example, which may be approved and/or signed at various levels of a
2 workflow by appropriate users. Such data presented for signature and
3 approval at the various levels is retrieved from dynamic database 36 in
4 the described configuration. Dynamic database 36 may be referred to
5 as a target application for implementation of system 10 according to the
6 present invention.

7 Dynamic database 36 is implemented as a relational database or
8 an object-oriented database in exemplary embodiments. Data is stored
9 within a plurality of interrelated portions, such as tables, columns, rows,
10 fields, etc. of dynamic database 36. A given user may have authority
11 to change and sign data in one or more portion but have restricted
12 access to other portions. Other users may authority to review, change
13 and sign data within all portions of dynamic database 36. Further,
14 digital data stored within at least some portions of database 36 may
15 dynamically change with respect to time (e.g., a given portion may
16 receive and store updated information or data over a given period of
17 time while another portion contains the same data over the same period
18 of time).

19 Snapshot broker 30 is configured to handle and manage requests
20 from client terminal 12. In one exemplary implementation, snapshot
21 broker 30 is provided as a server network domain. Such listens and
22 processes requests from digital signature library 38 within a given client
23 terminal 12. Snapshot broker 30 connects to dynamic database 36 to
24 gather data to be reviewed, signed and/or verified as described below.

1 In addition, snapshot broker 30 stores data sets of newly signed
2 snapshots (described below) within snapshot database 32 and digital
3 signatures which correspond to respective snapshots within approval
4 tables 34.

5 The depicted structural components of Fig. 1 and Fig. 2 are
6 exemplary. The indicated databases may be implemented within a single
7 storage device, or in additional multiple storage devices in other
8 embodiments. For example, approval tables 34 may be implemented as
9 a portion of dynamic database 36, or implemented as a distinct
10 database as shown in Fig. 2.

11 Snapshot database 32 is configured to store a series of snapshot
12 data records which comprise historical flat or static records (also
13 referred to as data sets) of possibly dynamic or changing digital data
14 stored within dynamic database 36. Snapshot database 32 may be
15 referred to as a static database. A user of client terminal 12 signs
16 such snapshots of digital data at recorded points in time.

17 Snapshot database 32 provides an audit record that can be flat
18 verified or cross-verified against current database entries of dynamic
19 database 36 as described below. The flat verification of snapshots
20 ensures that the static data of the snapshots and respective digital
21 signatures have not changed since they were originally signed and binds
22 the user to the data set they signed.

23 Cross-verification ensures that the data set of a given snapshot
24 matches a corresponding data set within dynamic database 36 (i.e., the

1 data from at least a portion of dynamic database 36 at an initial point
2 of time when the snapshot was digitally signed matches the data within
3 the corresponding portion of dynamic database 36 at a subsequent
4 moment in time). Corresponding portions of dynamic database 36 refer
5 to the same or matching tables, fields, rows, columns, etc. of dynamic
6 database 36 at different moments in time regardless of the data stored
7 in such portions. Cross-verification additionally verifies the data of
8 dynamic database 36 at the subsequent point in time against the digital
9 signature of the snapshot signed at the initial moment in time. Such
10 is described in detail below.

11 Snapshot database 32 may additionally store a list of certificate
12 authorities (such as CA server 14) which it trusts to authenticate users
13 who originally digitally signed the data snapshots using their
14 corresponding certificates. In the described arrangement, snapshot
15 database is implemented as a lightweight directory access database
16 (LDAP). Other arrangements are possible including implementing
17 snapshot database as separate tables within dynamic database 36
18 comprising a target application database.

19 Digital signature library 38 is implemented as instruction code
20 resident upon client terminal 12 for execution by processing circuitry 13.
21 Digital signature library 38 provides an interface between user interface
22 application forms from server 16 and digital signature components
23 including private key management system 18, snapshot broker 30, and
24 public key infrastructure (PKI) components such as certificate authority

1 server 14. Digital signature library 38 coordinates communications
2 intermediate servers 14, 16, private key management system 18 and
3 client terminal 12. In the described configuration, digital signature
4 library 38 utilizes TCP/IP communications. Other protocols are possible.

5 Approval tables 34 may be implemented as a separate database
6 from dynamic database 36 or as distinct tables within dynamic database
7 36. Approval tables 34 are utilized to store digital signatures of
8 respective snapshots during a given implementation of system 10.

9 Further, approval tables 34 store reference data regarding the
10 digital signatures and snapshots. In general, approval tables 34 provide
11 data and embedded procedures which bind flat historical snapshots with
12 current data in dynamic database 36. For example, approval tables 34
13 store reference data such as database query information or procedures
14 used to recreate respective snapshots of digital data stored within
15 snapshot database 32 using dynamic database 36 at subsequent moments
16 in time. Such query information can include identifiers for fields, rows,
17 columns, tables, etc. of dynamic database 36 and is stored when the
18 data set of the snapshot is digitally signed and referenced to the
19 respective snapshot. Alternatively, the query information may identify
20 a version controlled file which contains the database query procedures
21 if such procedures are extensive.

22 Such query information may be recalled at subsequent moments
23 in time to regenerate data from portions of dynamic database 36
24 corresponding to the stored snapshots (which were generated using data

1 within dynamic database 36 from corresponding portions at an initial
2 moment in time). Approval tables 36 also include identification
3 information (also referred to as fingerprint identification information)
4 identifying digital users or signers and their respective signed snapshots.
5 For example, an employee identification number may be utilized to
6 identify the signer of a snapshot or a fingerprint ID to identify the
7 digital certificate used for signing.

8 Approval tables 34 also include data regarding workflow (i.e.,
9 routing of signed data between different levels of authority for multiple
10 signatures) in the described embodiment. Approval tables 34 include
11 a snapshot procedure map or "trees of forms" to provide single, digitally
12 signable entities while removing variations in display formats (e.g.,
13 spaces, number of significant digits, etc.). Such reduces the chances
14 that cross-verification would fail inappropriately even though stored data
15 is correct.

16 Approval tables 34 also define how a given signature level
17 changes over time so application forms and schemas (within certain
18 constraints) can change over time but still preserve cross-verification of
19 stored historical digital signatures with respect to corresponding current
20 data within dynamic database 36 as described further below.

21 Such provides the ability to cross-verify actual current data of
22 dynamic database 36 with historical snapshots of corresponding portions
23 of dynamic database 36 while taking into account the fact that signature
24 hierarchy may or may not restrict changes to certain portions of

dynamic database 36 for a given signature or authority level in the workflow process.

As mentioned above, system 10 is configured to verify a data set of a snapshot stored at an initial moment in time against the digital signature of the snapshot (also made at the initial moment in time) at a subsequent moment in time (which may be days, months, years, etc. after the initial moment in time). Further, system 10 is configured to verify a data set present within a corresponding portion of the dynamic database 36 at a subsequent point in time, which also corresponds to the static or flat data set of the respective snapshot, against the digital signature of the snapshot made at the initial moment in time.

As generally used herein, an initial moment in time refers to a time when a snapshot is digitally signed by a user. Accordingly, the data set which is signed corresponds to data stored within at least a portion of dynamic database 36 at the initial moment in time. The snapshot of the data is stored in snapshot database 32 at this time. Further, the digital signature of the snapshot, identification information of the signer, and query information identifying corresponding portions of dynamic database 36 storing the snapshot data set are stored in approval tables 34 at the initial moment in time.

Subsequent moments in time occur after the initial moments in time and refer to periods of time wherein verification operations are performed as defined below. Verification operations include a first verification (also referred to as a flat verification) operation of verifying

1 a data set of a snapshot with the digital signature of the snapshot.
2 Such indicates whether the data set of the snapshot and the digital
3 signature are valid.

4 Verifying operations also include a second verification (also
5 referred to as cross-verification) operation of verifying a dynamic data set
6 (defined as data within dynamic database 36 at the subsequent moment
7 in time which corresponds to the data set of the snapshot signed at the
8 initial moment in time) with the digital signature of the snapshot. Such
9 indicates whether the dynamic data set of dynamic database 36 and the
10 digital signature are valid.

11 Accordingly, if the first verification operation is valid and the
12 second verification operation fails, it is known that the dynamic data set
13 of dynamic database 36 has been changed.

14 The discussion of verification operations now continues with
15 reference to depicted components of Fig. 2. Processing circuitry 13 is
16 operable to generate a digital signature by digitally signing a snapshot
17 data set extracted from dynamic database 36 (e.g., digital data within
18 at least a portion of dynamic database 36 such as desired fields, rows,
19 columns, tables, or any combination thereof) at an initial moment in
20 time using the user's private key. Such digital signature of the
21 snapshot includes time/date information as well as query information
22 which can be utilized to generate a corresponding data set from
23 dynamic database 36 at a subsequent moment in time.

1 The digital signature generated from the snapshot data set and
2 identification information regarding the signer (user) is stored within
3 approval tables 34 with references to one another. Further, query
4 information (e.g., queries applied to dynamic database 36) for generating
5 the data set corresponding to a given snapshot from dynamic database
6 36 is stored in approval tables 34 and referenced to the respective
7 snapshot. Such enables comparison of digital data from a given portion
8 (e.g., desired fields, rows, columns, tables, etc.) of dynamic database 36
9 at the initial moment of time with data within the corresponding portion
10 (e.g., fields, rows, columns, tables, etc. corresponding to the above given
11 portion) of dynamic database 36 at the subsequent moment in time.

12 For example, at such subsequent moment in time, the stored
13 query procedure is applied to dynamic database 36 to extract the data
14 set present within the corresponding portion of dynamic database 36 at
15 the subsequent moment in time which may have changed from the
16 initial moment in time. Such verifies the corresponding data set from
17 dynamic database 36 at the subsequent moment in time against the
18 corresponding digital signature of the snapshot.

19 In one exemplary method of verifying a data set of a snapshot
20 against the digital signature in accordance with first verification
21 operations, processing circuitry 13 of client terminal 12 is configured to
22 initially provide a first hash of the data set within the snapshot to be
23 verified from snapshot database 32. Processing circuitry 13 is also
24 configured to request retrieval of the digital signature which corresponds

1 to the snapshot to be verified from approval tables 34.

2 Processing circuitry 13 thereafter generates a second hash from the
3 digital signature. Such second hash is generated using the public key
4 corresponding to the private key of the user or other entity which
5 originally digitally signed the data set of the snapshot to provide the
6 digital signature corresponding to the snapshot to be verified. To
7 determine the appropriate public key, processing circuitry 13 retrieves
8 user identification information from approval tables 34 which also
9 corresponds to the snapshot to be verified. Such user identification
10 information is applied to certificate authority server 14 to extract the
11 desired public key.

12 Following generation of the first hash and second hash, processing
13 circuitry 13 compares the generated hashes to verify the data set of the
14 snapshot against the digital signature. Any failure of verification
15 indicates a change of the data set of the snapshot or the digital
16 signature. Otherwise, both the snapshot data set and corresponding
17 digital signature are valid.

18 As indicated, the present invention also provides verification of the
19 dynamic data set in accordance with second verification operations. The
20 first and second verification operations may be performed individually
21 or together dependent upon the data set being verified.

22 In one exemplary methodology, processing circuitry 10 operates to
23 extract query information from approval tables 34 utilized to extract the
24 dynamic data set from dynamic database 36 at a subsequent point in

1 time after the corresponding data set was digitally signed at the initial
2 moment in time to create the snapshot.

3 Using the appropriate query information, processing circuitry 13
4 extracts the data set within dynamic database 36 to obtain the dynamic
5 data set. Such extracted dynamic data set is hashed by processing
6 circuitry 13 to provide a third hash. Such third hash is compared with
7 the second hash (determined from the digital signature) to verify the
8 dynamic data set against the respective digital signature. No verification
9 indicates different data within the dynamic data set with respect to the
10 snapshot data set if the snapshot verification procedure (first verification
11 operation) from above was verified.

12 Referring to Fig. 3 - Fig. 6, exemplary methodologies of operation
13 of client terminal 12 are shown. The depicted methodologies described
14 further below may be implemented as software or firmware executable
15 instructions within client terminal 12 for execution by processing circuitry
16 13. Alternatively, such methodologies may be implemented as hardware
17 in another arrangement.

18 Referring initially to Fig. 3, a methodology 50 for implementing
19 digital signature operations of a snapshot data set is shown.

20 Upon execution of methodology 50, a user authentication
21 procedure is performed at step S10. A user, such as an individual,
22 may be required to input a log-on identification such as employee
23 number, name or other identifying information at step S10.

1 Following successful authentication, a user enters and or reviews
2 data presented in a form of a given application of application server
3 16 at step 12. Such can include a series of forms, checklists, etc.
4 Such data is either entered or reviewed with respect to dynamic
5 database 36.

6 At step S13, a user initiates a call for a digital or electronic
7 signature service. For example, an approve button may be provided on
8 an electronic form used to enter and or review data with respect to
9 application server 16.

10 At step S14, an operation is conducted to check the validity of
11 the user certificate. Such is typically performed with respect to
12 certificate authority server 14 utilizing certificates stored within certificate
13 database 20. The methodology 50 may be terminated if the user
14 certificate is not found to be valid at step S14.

15 Otherwise, a snapshot of digital data either entered or reviewed
16 via an application form is created at step S16. Such includes creating
17 a flat or static file of data being reviewed or entered by the user.

18 At step S18, client terminal 12 operates to digitally sign the
19 snapshot. Exemplary details of such digital signature of the data set
20 of the snapshot are described further with reference to methodology 52
21 of Fig. 4.

22 At step S20, client terminal 12 sends the data set of the snapshot
23 to application server 16 for storage. Snapshot broker 30 operates to
24 store the snapshot data set in snapshot database 32.

At step S22, snapshot broker 30 operates to store the corresponding digital signature, query information and user identification information within approval tables 34.

Referring to Fig. 4, methodology 52 depicts exemplary steps for digitally signing a created snapshot.

Initially at step S30, client terminal 12 digests the data set using a date/stamp as well as a snapshot procedure utilized to generate the snapshot. Such snapshot procedure can include query information of dynamic database 36 utilized to retrieve and generate the digital data within the stored snapshot. Step S30 can also be referred to as a hashing operation. Exemplary algorithms can be utilized to implement methodology 52 including MD5, MD4, SHA, etc.

At step S32, a personal identification number (PIN) or pass-phrase is utilized to unlock the private key of the user or signer.

A step S34, the hash created in step S30 is encrypted using digital signature operations and the user's private key to create the digital signature of the snapshot. Exemplary encryption algorithms include RSA, DSA, PGP, ECC, etc.

Referring to Fig. 5, an exemplary methodology for 54 performing snapshot or flat verification of a digital signature is illustrated. The depicted methodology 54 verifies that the data set and digital signature of such snapshot data have not been altered in any way since storage and creation of the snapshot.

1 In one embodiment of the invention, a digital certificate is not
2 necessary for a user who wishes to verify another signature using
3 methodology 54. A user with access to application server 16 can verify
4 another signature regardless if personally they have signature authority
5 through a certificate in one exemplary configuration.

6 At step S40, a user logs on to the client terminal 12 using
7 appropriate identification information.

8 At step S42, the user identifies a data set such as a snapshot to
9 be verified. Such can be requested utilizing lookup forms presented to
10 client terminal 12 from application server 16.

11 At step S44, static data is retrieved from snapshot database 32
12 which corresponds to the identified snapshot to be verified.

13 At step S46, the retrieved data from step S44 is digested in a
14 one-way hash operation.

15 At step S48, user identification information such as a signer
16 identification, certificate identification, certificate authority identification
17 is retrieved from approval tables 34 corresponding to the original signer
18 of the data set.

19 At step S50, the public key corresponding to the entity which
20 signed the snapshot is extracted from certificate database 20 of
21 certificate authority 14 using the identification information retrieved at
22 step S48.

23 At step S52, the stored digital signature which corresponds to the
24 retrieved snapshot data is retrieved from approval tables at step S34.

At step S54, the digital signature retrieved at step S52 is decrypted using the public key extracted at step S50 to produce a hash.

At step S56, the hash created at step S46 and the hash created at step S54 are verified in a comparison operation.

If the hashes do not verify, a data verification failure is displayed indicating a change in either the digital signature or the snapshot data at step S58.

Otherwise, if the hashes are compared and verified successfully, the snapshot data is indicated as being verified against the signature at step S60.

Referring to Fig. 6, an exemplary methodology 56 for performing cross-verification of an electronic or digital signature of snapshot data with respect to corresponding digital data at a subsequent moment in time (i.e., dynamic data set) within dynamic database 36 is illustrated.

In one configuration of system 10, no digital certificate is needed for a user to verify another's signature. Anyone with access to dynamic database 36 can verify another's signature regardless if they personally have signature authority through a certificate in such a configuration. Methodology 56 may be launched automatically at critical points in an application's approval process (e.g., before a next review/signing level) as defined by an application's workflow in an exemplary arrangement.

At step S70, a user authentication operation is performed wherein proper users log-on to system 10.

1 At step S72, the user selects the appropriate data set of a
2 desired snapshot to be verified.

3 At step S74, client terminal 12 retrieves information from approval
4 tables 34 which corresponds to the identified data set of step S72.
5 Such information can include an original date/time information utilized
6 at step S30 to create the snapshot, and dynamic database query
7 information utilized to retrieve dynamic data corresponding to the
8 snapshot from the dynamic database 36.

9 At step S76, the retrieved query information is utilized to recreate
10 the snapshot using current data stored within dynamic database 36.

11 The dynamic data set of information from dynamic database 36
12 which corresponds to the requested snapshot 32 is digested or hashed
13 by client terminal 12 at step S78.

14 At step S80, client terminal 12 retrieves identification information
15 regarding the user from approval tables 34. Such can also include
16 certificate and certificate authority identification information
17 corresponding to the signer of the snapshot data set.

18 At step S82, the public key of the user who originally signed the
19 snapshot data is extracted at step S82 from certificate authority server
20 14.

21 At step S84, client terminal 12 retrieves the digital signature
22 which corresponds to the snapshot from approval tables 34.
23
24

At step S86, a retrieved digital signature from step S84 is decrypted using the public key of the original signer. A decrypting operation creates a hash from the signature.

At step S88, the processing circuitry determines whether the hashes from steps S78 and S88 verify.

If not, a data cross-verification failure may be indicated at step S90 indicating the dynamic data set does not correspond to the data retrieved from the desired snapshot. Alternatively, such data cross-verification failure can also indicate a failed or corrupt digital signature retrieved from approval tables 34. However, if the flat verification of Fig. 5 is performed and verification is indicated, such would indicate corruption or other change within dynamic database 36.

Alternatively, if the condition of step S88 is affirmative, the digital signature of the snapshot data is verified against the dynamic data set of the dynamic database 36 as indicated at step S92.

In compliance with the statute, the invention has been described in language more or less specific as to structural and methodical features. It is to be understood, however, that the invention is not limited to the specific features shown and described, since the means herein disclosed comprise preferred forms of putting the invention into effect. The invention is, therefore, claimed in any of its forms or modifications within the proper scope of the appended claims appropriately interpreted in accordance with the doctrine of equivalents.